# Adaptive Simulated Pilot

Phillip D. Stroud (stroud@lanl.gov)
Los Alamos National Laboratory
LAUR-95-2879, April 18, 1997
(Shortened verson appears in AIAA Journal of Guidance, Control, and Dynamics, Vol 21, No. 2, 1998)

Abstract - A simulation actor is developed that emulates the behavior, including the ability to adapt to new conditions, of a human pilot of an airplane-based laser system. This flight controller performs the cognitive task of keeping the plane within a designated region of air-space while maximizing the performance of its laser coverage mission. The flight controller has been implemented as a parameterized production system. Because the system operates in a complex environment, it is not practical to pre-program an appropriate flight control behavior for every possible environmental condition. An evolutionary approach is used instead, in which the process by which the flight controller performs its cognitive task adapts to external conditions. This adaptive capability is provided by a cognitive architecture which uses a genetic algorithm to evolve a population of trial flight control processes, and an internal simulation to represent knowledge of the external environment and evaluate trial flight control processes. Human-like adaptation of flight control behavior is demonstrated for widely differing environmental conditions.

Nomenclature

$x, y$ = location in horizontal plane (km)
$b$ = airplane bearing, clockwise from threat direction, in $(-\pi,\pi)$
$R$ = minimum turning radius
$s$ = $\sin(b)$
$c$ = $\cos(b)$
$\sigma$ = sign(s); =1 when heading right, -1 when heading left
$F_T$ = Boolean flag indicating a request to turn forward
$Y_R$, $Y_L$ = y coordinate of offset line when heading right and left
$F_R$, $F_L$ = Boolean flags specifying whether the plane must turn forward at the right and left extremes of its orbit
$x_m$ = x coordinate location of center of allowed orbit box, km.
$x_h$ = half-width of allowed orbit box, km.
$x_L$ = $x_m - x_h$ = left-most x coordinate of flight path configuration, km.
$x_R$ = $x_m + x_h$ = right-most x coordinate of flight path configuration, km.
$b_{cut}$ = magnitude of maximum desired heading, radians.
$W$ = width of assigned Airborne Laser (ABL) air-space, km

# I. Introduction

There are many applications for machines that can perform cognitive tasks. One important application is simulation of complex systems that have intelligent elements in them.[1,2] In particular, battlefield simulation[3,4,5] (which is becoming an essential tool for training, operations planning, and technology development) requires software actors that can emulate human operators and decision makers. The battlefield environment is a complex system composed of many interacting elements, in which unpredictable collective phenomena emerge.[6] Humans or human organizations, by virtue of their intelligence, can adapt the processes by which they perform their assigned cognitive task, as unanticipated conditions develop.[7,8] A methodology is developed that enables software simulation actors, like the humans they emulate, to not only perform a cognitive task, but to adapt the process by which the task is performed to changing, un-preprogrammed external conditions.

A three part cognitive architecture is used to construct the software actors. The low-level cognitive part executes a process that accomplishes the assigned cognitive task. This cognitive process is implemented in a way that has a set of adjustable control parameters, so that a great variety can be produced in the cognitive process by changing the parameter values. The behavior of human operators or experts can be mimicked by the software process over some domain of environmental conditions by setting appropriate values of the control parameters (i.e. by supervised training or knowledge engineering). Alternative behaviors, represented by alternative control parameter values, might be more suitable as the environmental conditions change. The space of all possible environmental conditions is too large to pre-program optimal control parameter settings for every contingency. However, an efficient genetic algorithm can be used to adapt the control parameters as novel conditions emerge. The second part of the cognitive architecture is an evolutionary apparatus. It uses a chromosome (binary string) representation of the set of cognitive process control parameters. It generates and maintains a population of chromosomes, and applies the genetic algorithm to continuously search for processes superior to the current low-level cognitive process. The third element of the cognitive architecture is a mechanism for estimating the fitness of trial sets of process control parameters. This mechanism uses an internal simulation of the external system. This three part cognitive architecture is described in section II.

The flight controller of an airborne laser theater missile defense (ABL-TMD) system is used as a test-bed for developing and demonstrating the approach.[9,10] The flight controller (besides issuing steering command signals) performs the cognitive task of selecting flight paths that satisfy several conflicting objectives. First, it has to keep the aircraft within a designated rectangular region of air space. If the plane ventures through the front of this assigned box, the ABL becomes more vulnerable to attack. The sides of the box are set when flight corridors are reserved for other air operations. As shown in Fig. 1, the field of fire of an ABL does not extend completely aft, because of the physics of the air flow boundary layer, and the geometry of the plane. The second objective of the flight controller is to keep the field of fire over the launch zone to the greatest extent possible. The ABL can turn to engage targets that are launched behind the field of fire. The flight controller has to decide when to accept a request to turn towards a target. This capability prevents the use of simple scripted flight controllers, because the need to turn to a target can not be predicted. Another objective is to keep the plane as close to the threat area as possible, because the deliverable intensity drops off with range. A parameterized production system has been developed to perform this cognitive task. A great variety of flight path configurations can be obtained by varying the control parameter values. This production system is described in section III.

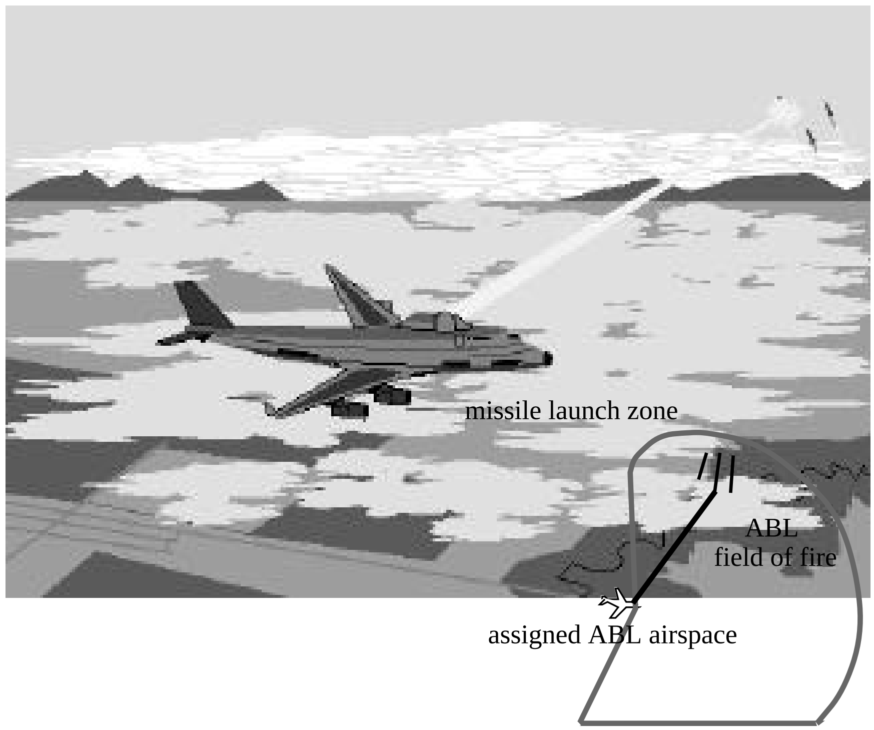


Fig. 1. The airborne laser flight controller has the cognitive task of selecting flight paths that maximize coverage of the threat zone while staying in assigned air-space.

Sections IV and V describe the internal simulation used to estimate the fitness of trial controllers and the evolutionary apparatus. Section VI presents a demonstration of how the flight path selection process can adapt to un-preprogrammed battlefield conditions, leading to substantially improved performance.

## II. Cognitive Architecture.

Research in artificial intelligence has identified many of the essential elements (working memory, sensors, knowledge representation, evaluators, manipulators) of systems that perform cognitive tasks, and various ways of combining them into cognitive architectures.[11,12,13] Machines have been constructed to perform such cognitive tasks as problem solving, language interpretation, image processing, logical inference and deduction, search, and learning.[14] A three-part cognitive architecture has been developed to enable a simulation actor to adapt its cognitive process to changing, unanticipated conditions in the system. The architecture is based on two principles: knowledge about the external environment can be represented within the actor by an internal simulation[15], and an actor can adapt its behaviors by using an internal evolutionary process.[16]

A schematic of this architecture is shown in Fig. 2. In simulation of complex systems composed of interacting elements, the interacting elements are simulated with software actors which interact through asynchronous message passing. These simulation actors have a collection of data, a set of methods or actions by which they manipulate data and interact with the rest of the system, and a cognitive segment. The actor's cognitive segment has a software implementation of the process by which the cognitive task is accomplished. It also has a higher level mechanism that can adapt the low-level process to changing environmental conditions. In non-adaptive actors, the cognitive segment consists only of the cognitive task process, which, along with the actor's physical state and collection of actions, forms the actor. The process relating actions to physical state can be implemented as a problem solving or search algorithm, a control system, a simple set of rules, an expert system, or a neural network of various configurations. Whatever the implementation of the process, there will be a set of adjustable control parameters that determine the behavior (e.g. the set of connection weights for a neural network representation). In non-adaptive actors, the adjustable control parameters that control the actor's cognitive processes are pre-programmed (or pre-trained) to build in expertise. The high-level component of this cognitive architecture provides a mechanism for an actor to adapt its low-level cognitive process by adjusting the control parameters.

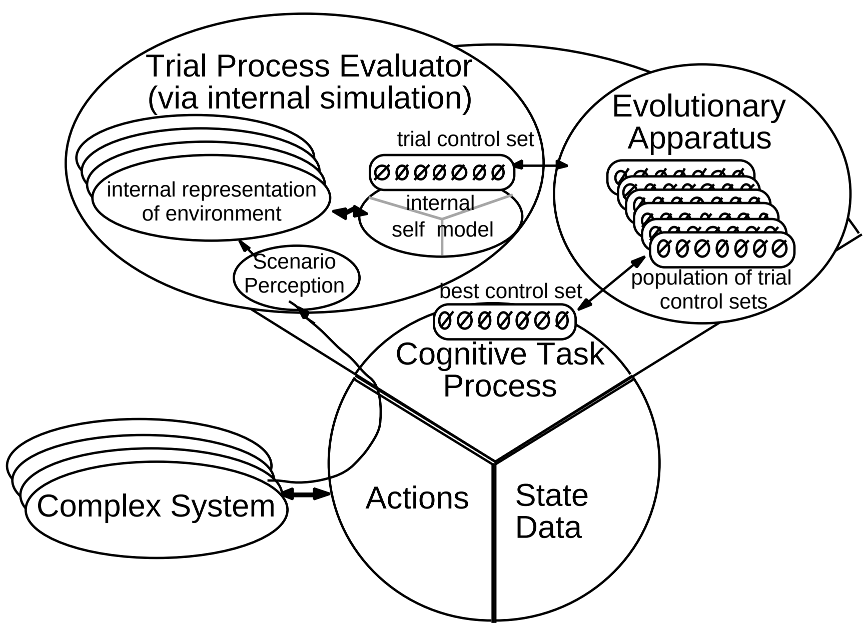


Fig. 2. Cognitive architecture for constructing adaptive actors. The process by which the actor performs its cognitive task is specified by a set of process control parameters. An evolutionary apparatus generates and evolves a population of trial control parameter sets. An internal simulation, representing knowledge of the external system, provides fitness evaluations of trial control parameter sets.

Two primary elements make up the high-level cognitive part: a mechanism that can estimate the effectiveness of trial behaviors using an internal representation of the external world, and an evolutionary apparatus. The internal representation is tied to currently perceived external conditions. The internal representation is based on an internal simulation of relevant parts of the external environment. This internal simulation is the mechanism by which the actor incorporates knowledge about the external world.[15] The internal simulation generates (and stores in a working memory) random missile launches that are consistent with the currently perceived battlefield conditions (expected launch zone, current atmospheric conditions, expected opposing forces tactics, etc.), which are stored as part of the actor's physical state data. The perceived external conditions can be changed by direct sensors, or by receipt of intelligence updates. There is also an internal model of the ABL system, including its flight path selection process, and a means of evaluating how well a trial process (represented by a trial set of control parameters) would perform against the missiles in memory. Since the fitness of a given process will have a large stochastic component, the internal fitness estimation must evaluate a trial behavior over many possible sorties to obtain a valid fitness estimate. In the case of the ABL, the fitness estimation requires evaluation of about 10,000 missile engagements.

The evolutionary apparatus generates and maintains a population of alternative process control parameters, in a working memory. It uses a genetic algorithm to evolve this population. The behavior produced by the set of control parameters is encoded into a chromosome. The population of chromosomes is evolved by the evolutionary apparatus, using genetic cross-over, mutation, and fitness based selection. Fitness based selection drives the population to better adaptations. The evolutionary apparatus uses the internal simulation to evaluate various trial behaviors. When a better performing cognitive process is found, it replaces the current cognitive process.

## III. The Rule-based Flight Controller.

A flight controller can be seen as operating on two levels. At the lower level, the flight controller issues aircraft steering control signals. It provides the steering corrections necessary to keep the aircraft on a particular course or maneuver. On the higher level, the flight controller plans a flight path that enables the system to best perform its mission. In selecting the path configuration, the controller is guided by knowledge regarding such factors as the ideal distance to maintain behind the front of the box, whether to allow turns away from the threat, how far to either side to go before turning around, and when to ignore requests to turn toward a target. The flight controller can not simply be a scripted orbit, because the plane would then be unable to react to the threat. A simulated ABL has been incorporated into the Theater Air Command and Control Simulation Facility (TACCSF).[17] This simulated ABL employs a human pilot to manually fly the plane in a flight simulator. A flight path selection process has been developed to emulate the behavior of human pilots. The flight controller is implemented as a production system (a collection of IF-THEN production rules) that maps any input aircraft state into a turn signal output. The production system is parameterized, and the values of the parameters determine how the controller performs the higher-level path selection task.

For constant altitude flight, the plane's state is described by three parameters: x and y, which give the location, and b, the relative heading. The coordinate system is defined relative to the assigned ABL air-space as shown in Fig. 3. The forward-most extent of the allowed region of airspace is designated as y=0. Presumably, the aircraft takes on unacceptably high risks if it passes forward of this line. The maximum allowed width, W, of the assigned airspace is allocated along with air corridors for other air operations associated with the regional conflict. The coordinate system origin is located at the center of the front of this maximum width box. The y direction is normal to the front of the box, toward the threat. The positive x direction is to the right when facing the threat. The relative heading is the clockwise angle from the positive y direction to the direction of the plane's velocity. In addition to these three plane state variables, the flight controller has a Boolean input parameter, $F_T$, that indicates whether or not the fire controller has requested that the plane turn forward to bring a target into the field of fire.

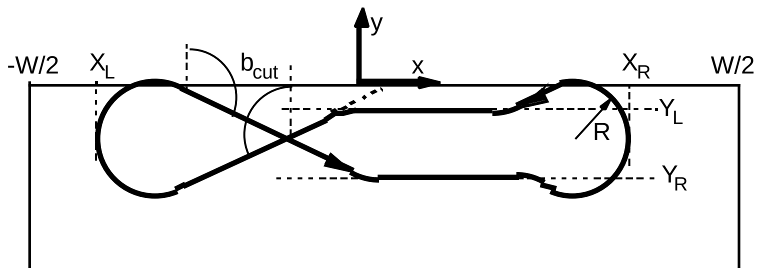


Fig. 3. Flight path geometry. The width of the assigned air-space is W, and the origin of the local coordinate system is at the center of the front of the assigned air-space.

The flight controller output can be a real-valued turn rate (in degrees per second) ranging continuously from a maximum left turn to a maximum right turn. For simplicity, the flight controller signal can be discretized to two possible values: maximum right turn and maximum left turn. Straight line flight is obtained by alternating control signals. For a plane speed of 0.25 km/s and a minimum turning radius, R, of 24 km, the path obtained by alternating maximum left and right turns every 2 seconds deviates about a straight line by only about one meter. The turn signal can also be represented as forward or backward (towards or away from the threat). If s>0, forward is to the left.

The production system consists of an ordered set of productions, of the form IF(condition) THEN (turn t), where t is left or right, and condition is a logical relation among the controller input variables. In the case that two or more productions have their conditions satisfied, the production earlier in the set is invoked. This is accomplished by making the first production an IF, the second through the second last ELSE-IF's and the last production an ELSE. This mechanism eliminates the need for explicit conflict resolution.

The first production (A) makes the plane fly back if it ever finds itself out in front of the box. Productions B and D prevent the plane from crossing the front of the box, while productions C and E prevent the plane from flying out across the side of the box. If the plane happens to be past the side of the box, they also get the plane back into the box. This behavior is obtained by making a rearward turn if not doing so would cause the plane to cross the box front (B). However, when the plane is near one of the front corners of the box, the rearward turn must be performed early enough to prevent crossing the side of the box (C). A forward turn is made if not doing so would cause the plane to cross the box side (E). However, when the plane is near one of the front corners of the box, the forward turn must be performed early enough to prevent crossing the front of the box (D).

Productions F and G prevent the plane from crossing a side of the box when the plane is heading rearward. If possible, the plane will execute a forward turn, but if necessary, it will execute a rearward turn. Production H causes the plane to turn forward whenever the heading is rearward of a specified maximum heading. The next pair of productions, I and J, forces the plane to approach the corners from far enough behind the front of the box so that there is enough room for a forward turn. This is accomplished by keeping the plane behind a diagonal line tangential to a turning circle in the corner, that has a direction given by the specified maximum rearward heading. Production K then checks to see if the fire controller has requested that the plane turn towards a target to bring it into the field of fire. Productions L and M get the plane to fly along lines parallel to the front of the box by turning rearward if ahead of these lines. There are two of these offset lines, one when heading left and one when heading right. Production N ensures the transition from the left side to the right side behavior. Finally, the last production, which is invoked when none of the others apply, turns the plane toward the threat. The 15 productions are as follows:

A: IF $y \geq 0$ THEN turn rearward.

B: ELSE IF ($c \geq 0$ AND $y \geq R(|s|-1)$ AND $\sigma(x-x_m) \leq x_h-R(1+c)$ ) THEN turn rearward

C: ELSE IF ($c \geq 0$ AND $\sigma(x-x_m) \geq x_h-R(1+c)$ AND $\sigma(x-x_m) \leq y+x_h+R(|s|-c)$) THEN turn rearward

D: ELSE IF ($c \geq 0$ AND $y \geq -R(|s|+1)$ AND $\sigma(x-x_m) \geq y+x_h+R(|s|-c)$) THEN turn forward

E: ELSE IF ($c \geq 0$ AND $y \leq -R(|s|+1)$ AND $\sigma(x-x_m) \geq x_h+R(c-1)$ ) THEN turn forward

F: ELSE IF ($c<0$ AND $\sigma(x-x_m) \geq x_h-R(c+1)$ ) THEN turn forward

G: ELSE IF ($c<0$ AND $y \leq -R(|s|+1)$ AND $\sigma(x-x_m) \geq x_h+R(c-1)$ ) THEN turn forward

H: ELSE IF ($c<\cos(b_{cut})$ ) THEN turn forward;

I: ELSE IF ($s \geq 0$ AND $F_R = 1$ AND $y \sin(b_{cut})>(R-x_m-x_h+x) \cos(b_{cut})+R \cos(b_{cut}-b)-R \sin(b_{cut})-2R$) THEN turn rearward;

J: ELSE IF ($s<0$ AND $F_L = 1$ AND $y \sin(b_{cut})>(R+x_m-x_h-x) \cos(b_{cut})+R \cos(b_{cut}+b)-R \sin(b_{cut})-2R$) THEN turn rearward;

K: ELSE IF ($F_T = 1$) THEN turn forward;

L: ELSE IF ($s \geq 0$ AND $y > Y_R-R(1-|s|) \operatorname{sign}[\cos(b_{cut})]$ ) THEN turn rearward;

M: ELSE IF ($s < 0$ AND $y > Y_L-R(1-|s|) \operatorname{sign}[\cos(b_{cut})]$ ) THEN turn rearward;

N: ELSE IF ($c > 0.995$) THEN turn rearward;

O: ELSE turn forward;

This production system generates sensible flight paths for small time steps. By adding small increments to some of the production conditions, the system can generate sensible flight paths for relatively large time steps.

There are several parameters that appear in the productions, that to a large extent determine the overall configuration of the resulting flight paths. In particular, the seven parameters $Y_R$, $Y_L$, $F_R$, $F_L$, $x_m$, $x_h$, and $b_{cut}$, can be varied to produce a great variety of flight path configurations. For example, the parameters $\{Y_R$=48km, $F_R$=1, $b_{cut}$ =105°, $x_h$=120km, $x_m$=0, $Y_L$=48km, and $F_L$=1$\}$ produces a figure-eight orbit in which the plane never turns away from the threat. The production system using this set of parameters is designated flyConE. A similar orbit, generated by a script, has been used in several major missile defense simulations[18], so flyConE can be considered to represent a generic expert ABL flight controller.

## IV. Simulation Based Performance Evaluation

The simulation package ELASTIC (Evolutionary Los Alamos Simulation-based Training for Intelligent Controllers) has been developed to provide a flexible capability to simulate the operation of various ABL flight controllers in a theater simulation environment. The package provides a collection of C++ classes and functions to construct actors that simulate the ABL aircraft, theater ballistic missiles, a missile tracker, a fire controller, a flight controller, and the laser beam. A perceived scenario describes the expected number of missiles, the launch timeline, and the geometry of the launch zone. The perceived scenario is updated to match currently perceived external conditions.

A sortie class runs a simulation of the ABL against a particular random set of missiles. As each missile is launched, it enters the track file, which is a data structure belonging to the flight controller. The missile positions are updated during their boost phase with a 4th order Runge-Kutta integration[19] of the rocket equation. An adaptive time step is used during the internal simulation. A fire controller determines which missile is selected from the track file, and when it is fired at. A high fidelity laser propagation model (including active compensation of the effects of atmospheric turbulence[20,21]) is used to calculate the intensity delivered to the missile. If a lethal fluence is delivered to the missile before burn-out, the missile is destroyed. The burn-out time of a given missile depends on its launch and aim point locations (missile range is controlled by thrust termination). In the case that there are more than one missile to shoot at, the fire control strategy is to select the target which can be destroyed in the least estimated amount of time. There is a chance that a missile will burn out while being illuminated, in which case it achieves its ballistic trajectory and leaks through the ABL missile defense layer.

The measure of performance or fitness is the fraction of missiles intercepted by the ABL, or the intercept probability. This intercept probability depends on many factors in addition to the flight controller. Some of these factors characterize the laser platform performance: laser power, beam aperture, laser wavelength, beam jitter, atmospheric turbulence compensation system performance, platform altitude and speed, maximum bank angle, turret slew rate. The laser system parameters have been arbitrarily selected to provide approximately a 50% intercept probability. The level flight minimum turning radius is 24 km. Another set of factors characterize the scenario, and may change drastically during the course of a regional conflict: the size and location of the threat or launch zone, the size and location of the protected asset zone, the missile types, number of launches, and timeline of launches.

The number of missile engagements or salvos that must be simulated to assess a flight control strategy is determined by the required accuracy in the estimator of the missile intercept probability, $P_I$. The error in the estimated $P_I$ is estimated with the (unbiased) standard deviation consistent with the total number of simulated intercepts, K, and launches, L:[19]

$$P_I = \frac{K}{L} \pm \sqrt{\frac{1}{L-1}\frac{K}{L}\left(1-\frac{K}{L}\right)}.$$

In a scenario where the individual missile kill probability is about 50%, on the order of ten thousand engagements are required to obtain a kill probability estimate that is valid to within 0.5%. The ABL actor has been implement on a Macintosh PowerPC. It takes 54 seconds to simulate 10010 missile engagements. An implementation onto a DEC AlphaStation runs approximately ten times faster. The simulation uses a 0.1 second time step while the ABL is engaging a missile. The simulation could be sped up by increasing this time step, at the expense of fidelity.

## V. Evolutionary Apparatus

The flight path configurations generated by the production system depend on seven parameters. These seven parameters can be represented as a string of bits, or chromosome. A seven bit gene can represent each of the five real-valued parameters, allowing 128 possible values for each. The two Boolean flags can be represented with one bit each. The seven production system control parameters can thus be represented as a string of 37 bits. Each of the $2^{37} = 1.37(10)^{11}$ different 37 bit strings represents a different set of control parameters, and a different strategy for generating a flight path. The chromosome is obtained by concatenating the binary representation of the following seven genes:

$$g_0 = 127\, Y_R \;/\; 75$$
$$g_1 = F_R$$
$$g_2 = 127\,(b_{cut} - 91°)/88$$
$$g_3 = 127\,(2\,x_h - 2\,R) \;/\; (W - 2\,R)$$
$$g_4 = 127\,(x_m + W/2 - x_h)/(W - 2x_h)$$
$$g_5 = 127\, Y_L \;/\; 75$$
$$g_6 = F_L$$

The flag for a forward turn at the right edge of the box and the y-offset of the right-heading leg are located near each other in the chromosome, as are other closely related data pairs. The set of control parameters that represents the phenotype of flyConE, {$Y_R$=48km, $F_R$=1, $b_{cut}$=105°, $x_h$=120km, $x_m$=0, $Y_L$=48km, and $F_L$=1} gives the set of seven genes {81, 1, 21, 54, 63, 81, 1}, which in turn gives a chromosome 1010001:1:0010101:0110110:0111111:1010010:1. The colons delineating the genes are for readability only. A chromosome is converted back to control parameter values with

$$Y_R = 75\, g_0 \;/127$$
$$F_R = g_1$$
$$b_{cut} = 91° + 88\, g_2/127$$
$$x_h = R + g_3\,(W/2 - R)/127$$
$$x_m = -W/2 + x_h + g_4\,(W - 2x_h)/127$$
$$Y_L = 75\, g_5/127$$
$$F_L = g_6$$

The collection of all possible behaviors (represented by all possible control parameter settings) evaluated against all possible environments consistent with perceived conditions, provides a well-posed problem for the genetic algorithm.[22] A gradient decent search strategy does not work because of the existence of many local optima. A simple form of genetic algorithm is used to search for better-adapted process control parameters as follows:[23,24] First, the internal simulation used to evaluate trial control parameters is tied into the currently perceived external conditions by setting the scenario data. An initial population of trial chromosomes is constructed by copying the original (current best) chromosome, and mutating a few bits at random (with a 15% bit flip probability). A population size of 36 chromosomes is used. The fitness of each of these trial chromosomes is then estimated, by simulating the performance of their corresponding flight control behavior. The population is then sorted by estimated fitness. The next step is to evolve the population by stepping through generations, looping over the four processes of 1) select parents from the population, based on their estimated fitness 2) breed new trial chromosomes, using genetic cross-over and mutation operations, 3) estimate the fitness of the trial chromosomes, and 4) sort new chromosomes into the population, replacing less fit chromosomes. The mechanism of punctuated equilibrium has been found to be effective: the mutation rate starts out at a large value, and is decreased after each generation using a power law annealing schedule. After evaluating a few hundred trials, the mutation rate is reset to the initial high value for another round of simulated annealing. Whenever a chromosome is found that has better estimated fitness than the current process, the control parameters are replaced with the better values.

## VI. Results

Scenario A.

In the first scenario, scenario A, the missiles are launched from a rectangular launch zone that extends from 200km to 400km in front of the assigned ABL air-space, and has a width of 400km. A scenario A sortie has 110 missiles launched over a 2.5 hour period. In this scenario, it is rare for there to be more than one engagable missile at a time. Each launch occurs at a random location within the launch zone, and at a random time. The assigned ABL air-space has a width of 500 km, and is centered left-to-right on the threat. The missile trajectories carry them to a range that is uniformly distributed on the interval from 400 to 600km. The general direction of the missile flight is toward the ABL zone (i.e. the negative y direction), but there is a ±25 degree spread in missile launch direction. The geometry of this scenario is shown in Fig. 4.

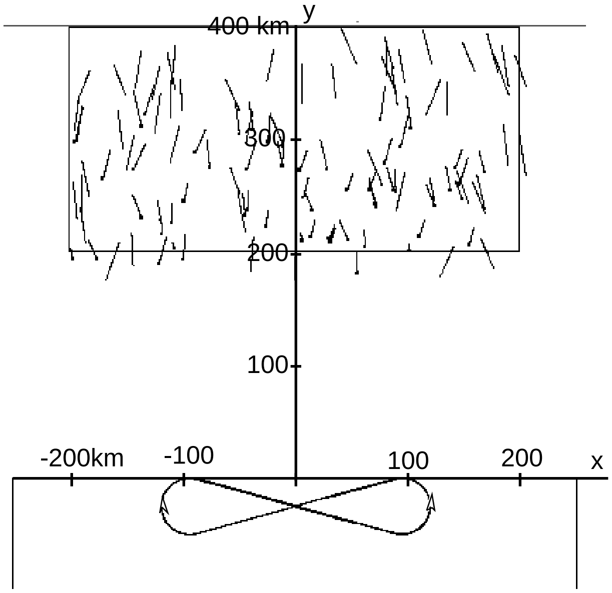


Fig. 4. Scenario A, and the flight path generated by the expert flight controller, flyConE.

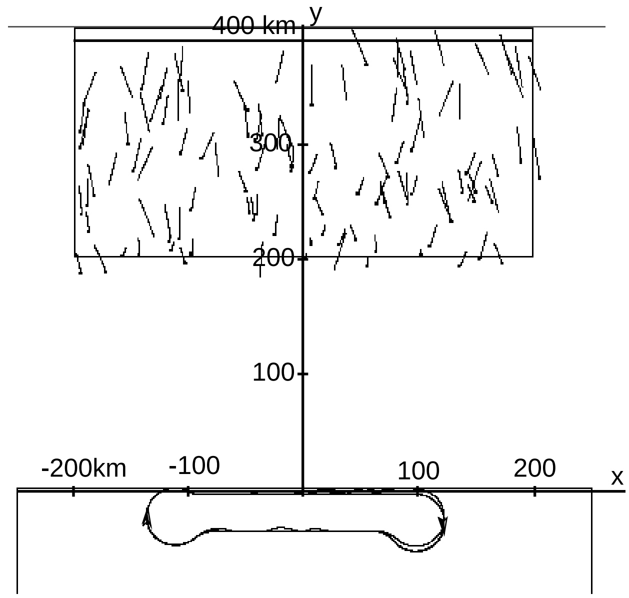


Fig. 5. The flight path generated by the flight controller that has adapted itself to scenario A.

The flyConE parameter set gives a flight controller that generates a figure-eight path that makes both turns toward the threat, positioned as far forward as possible in the assigned ABL air-space. The figure-eight is contained in a 240 by 48 km rectangle, which is centered left-to-right at the center of the threat launch zone. This flight controller is evaluated against scenario A, as shown in Fig. 4. The effectiveness of this flight controller is assessed by running 91 simulated sorties, each of which has 110 missile launches consistent with scenario A. This expert flight controller allows the ABL to intercept 5426±50 of these 10010 simulated missiles, giving an intercept probability of $P_I$=54.2% , and a leakage fraction of $P_L$=45.8%.

The control parameters are then allowed to evolve, using the evolutionary apparatus and simulation-based fitness evaluation with scenario A. After evaluating 544 trial control parameter sets, a new flight controller is obtained that intercepts 6235 of 10010 missiles. The leakage fraction has been significantly reduced from 45.8% down to 37.7%. This new flight controller, FlyConA, generates a clockwise racetrack configuration, and is obtained from the production system by using the parameter set {$Y_R$=5.9km, $F_R$=0, $b_{cut}$ =124°, $x_h$=129km, $x_m$=-9km, $Y_L$=38.4km, and $F_L$=1}. The flight path generated by FlyConA for the first of 91 simulated sorties is shown in Fig. 5. The actor has discovered that a racetrack configuration outperforms the expert figure-eight controller. It also has found that the clockwise racetrack should be offset to the left, so that the turn away from the threat occurs closer to the center of the threat than the turn into the threat. It has also adjusted the y-offset of the leftward and rightward heading legs to balance the needs to be closer to the threat, to have room to turn to the target, and to avoid having to turn backwards to make the turns at the ends of the racetrack.

Scenario B.

The scenario is then changed by decreasing the depth of the launch zone, so it extends from 200 to 300 km from the front of the ABL box. This new scenario, scenario B, might result from an opposing force strategy to reach deeper targets. When the expert figure-eight flight controller, flyConE, is used against scenario B, 7036 of 10010 missiles are intercepted, giving a leakage fraction of 29.7%. When flyConA, (the racetrack flight controller adapted to scenario A) is used against scenario B, 7978 of 10010 missiles are intercepted, for a leakage fraction of 20.3%. Some of the knowledge acquired from scenario A by flyConA transfers over to scenario B.

The control parameters are then evolved, using the evolutionary apparatus and simulation-based fitness evaluation with scenario B. After evaluating 644 trial control parameter sets, a new flight controller is obtained that intercepts 8646 of 10010 missiles. The adapted flight controller has reduced the leakage fraction to only 13.6%. This new flight controller, flyConB, is specified by the parameters {$Y_R$=61km, $F_R$=1, $b_{cut}$ =113°, $x_h$=139km, $x_m$=0, $Y_L$=61km, and $F_L$=1}. The flight path generated by flyConB for the first of 91 simulated sorties consistent with scenario B is shown in Fig. 6. This flight controller has discovered that when most of the targets are going to be within range, coverage can be increased by pulling back a little from the front of the box.

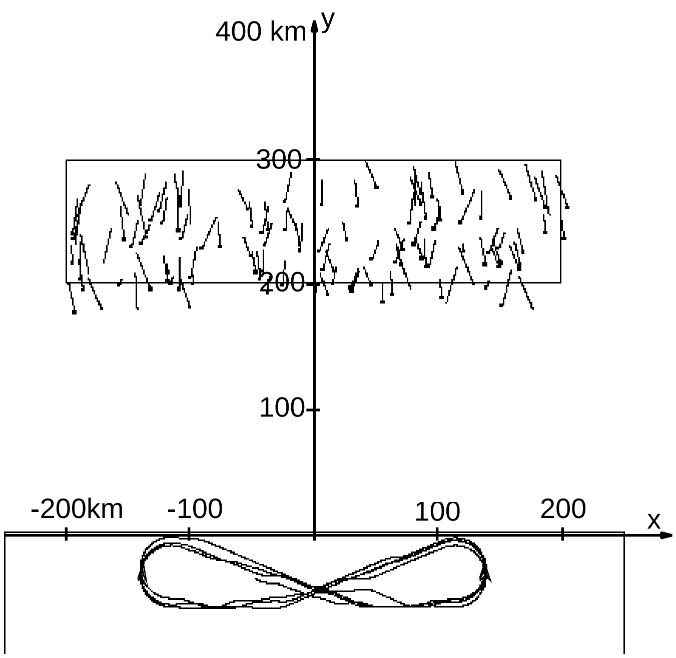


Fig. 6. Scenario B, and the flight path generated by a flight controller that has adapted its flight path selection process to scenario B.

Scenario C.

A third scenario, scenario C, has two 40 by 40 km launch zones, 300 km apart. The first is located 300 km in front of the ABL's assigned air-space, while the second is located 200 km in front. 60% of the missiles are launched from the second zone. The geometry is shown in Fig. 7. When the expert controller flyConE is used against scenario C, 5719 of 10010 missiles are intercepted, giving a leakage fraction of 42.9%. When flyConA, (the racetrack flight controller adapted to scenario A) is used against scenario C, 6810 of 10010 missiles are intercepted, for a leakage fraction of 32.0%. When flyConB is used against scenario C, 6568 of 10010 missiles are intercepted, for a leakage fraction of 34.4%.

The control parameters are then allowed to evolve, using the evolutionary apparatus and simulation-based fitness evaluation with scenario C. After evaluating 566 trial control parameter sets, a new flight controller is obtained that intercepts 7290 of 10010 missiles, reducing the leakage fraction to 27.2%. This new flight controller, flyConC, is specified by the parameters {$Y_R$=23km, $F_R$=1, $b_{cut}$ =120°, $x_h$=131km, $x_m$=-5km, $Y_L$=5.9km, and $F_L$=1}. The flight path generated by flyConC for the first of 91 simulated sorties is shown in Fig. 7. This flight controller generates a relatively complicated, asymmetric figure-eight configuration. It is easy to imagine unlimited variations on scenario C, with various proportions of the missiles launched from each box, various launch box locations and sizes. Each variant scenario would lead to a best adapted flight controller. This approach of adapting to any given scenario is more practical than the approach of pre-programming a flight controller that can generate appropriate flight path configurations for any possible scenario.

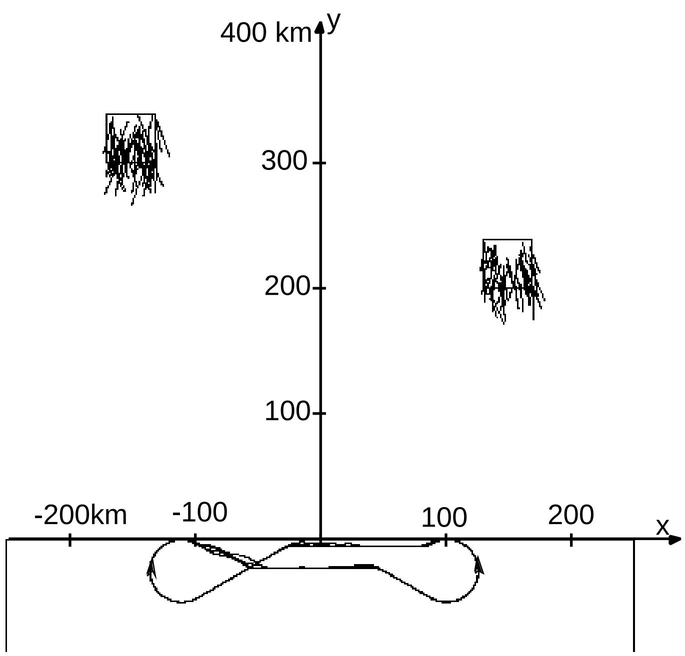


Fig. 7. Scenario C, and the flight path generated by a flight controller that has adapted its flight path selection process to scenario C.

Scenario D.

A fourth scenario, scenario D, has one 50 by 50 km launch zone that is located 100 km to the left of the assigned ABL air-space (i.e. centered at x,y = -350,250). The geometry is shown in Fig. 8. This scenario might occur when an ABL, based over one country, is protecting a second country from ballistic missiles launched from a third country. When the expert controller flyConE is used against scenario D, 4042 of 10010 missiles are intercepted, giving a leakage fraction of 59.6%. The other flight controllers give similarly poor performance, primarily because they are flying the plane in the wrong place.

The control parameters are then allowed to evolve, using the evolutionary apparatus and simulation-based fitness evaluation with scenario D. After evaluating 604 trial control parameter sets, a new flight controller is obtained that intercepts 8201 of 10010 missiles, dramatically reducing the leakage fraction to 18.1%. This new flight controller, flyConD, is specified by the parameters {$Y_R$=56.7km, $F_R$=0, $b_{cut}$ =173.5°, $x_h$=57km, $x_m$=-89km, $Y_L$=44.9km, and $F_L$=0}. The flight path generated by flyConD for the first of 91 simulated sorties is shown in Fig. 8. This flight controller flies a small angled racetrack in the front left corner of its assigned air-space. It is likely that a human pilot would develop a similar behavior.

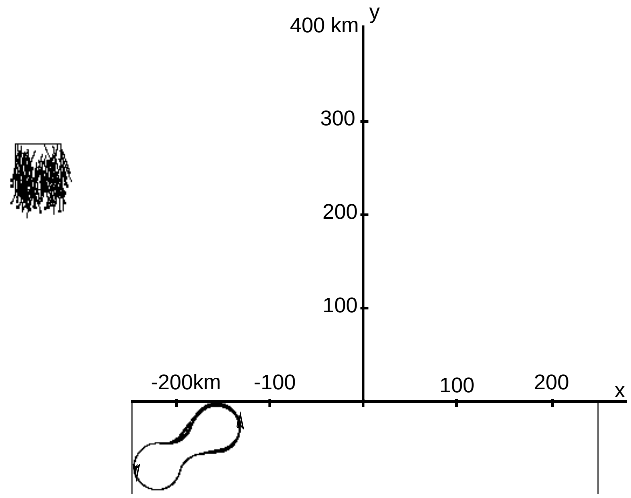


Fig. 8. Scenario D, and the flight path generated by a flight controller that has adapted its flight path selection process to scenario D.

## VII. Conclusion

A cognitive architecture has been presented for constructing simulation actors with the ability not only to perform cognitive tasks, but also to adapt the processes by which these tasks are performed, in response to changing environmental conditions. The implementation of the cognitive process as a parameterized production system is seen to be an effective way to represent a vast array of alternative behaviors. It can easily be set to mimic the behavior of a human operator over a limited domain. In addition, any knowledge obtained during the evolutionary adaptation can be readily extracted from the new control parameter values. The genetic algorithm is seen to be an effective method for evolving a population of trial process control parameters in a continuous search for better adapted processes. The use of an internal simulation is seen to be a practical way to represent knowledge about the external world.

An adaptive airborne laser flight controller demonstrated uncannily human-like un-preprogrammed adaptations when faced with several novel environmental conditions. It was able to discover a strategy of increasing the ABL's effective angular coverage by remaining further back than necessary when most of the missiles are within range. This strategy seems sensible in hindsight, but human pilots had simply never seen this domain. Had human pilots seen a similar scenario, they might well have adapted their flight path selection process in the same way. The adaptive airborne laser flight controller also produced processes suitable to an asymmetric two-launch-zone scenario, and to a scenario with a single launch zone off to one side. In all cases, the adaptive controller was able to generate behaviors that were significantly better than the expert behavior, or behaviors that were adapted to other scenarios. Like a human pilot, an ABL flight controller simulation actor that adapts its flight path selection process can significantly reduce the fraction of missiles that leak through.

## References.


1 J. M. Epstein, R. Axtell, Growing Artificial Societies: Social Science from the Bottom Up, Brookings Inst., (1996).

2 R. J. Gaylord, P. R. Wellin, Computer Simulations With Mathematica: Explorations in Complex Physical and Biological Systems, Springer Verlag, (1995).

3 Military Simulations Markets, Janes Information Group, ISBN: 0710612125 (1996).

4 P. P. Perla, The Art of Wargaming : A Guide for Professionals and Hobbyists, United States Naval Inst., ISBN: 0870210505, (1990)

5 C. R. Karr, D. Reece, R. Franceschini, "Synthetic Soldiers," IEEE Spectrum, vol. 34, no. 3 (1997).

6 M. M. Waldrop, Complexity, Simon & Schuster, NY , (1992).

7 J.W. Payne, J. R. Bettman, E. J. Johnson, The Adaptive Decision Maker, Cambridge University Press, (1993).

8 L. Bainbridge, "The Change in Concepts Needed to Account for Human Behavior in Complex Dynamic Tasks," IEEE Trans. Sys.t., Man, Cybern. A, Vol 27, No.3, pp 351-359, (1997)

9 G. Danczyk, S. Mortenson, W. Sailor, P. Stroud, "Airplane Based Free Electron Laser Concept Overview," proc. PBFEL Workshop, Los Alamos National Laboratory, (Dec. 4, 1990).

10 "Teams to Submit Bids For Airborne Laser," Aviation Week & Space Technology, (June 10, 1996).

11 H. Simon, The Sciences of the Artificial, MIT Press,Cambridge, Mass. (1981)

12 A. Newell, Unified Theories of Cognition, Harvard University Press, Cambridge, Mass, (1990).

13 M. Minsky, The Society of Mind, Simon & Schuster, (1987).

14 P. H. Winston, Artificial Intelligence, Addison-Wesley Publishing Company, Reading, MA, (1984)

15 H. Plotkin, Darwin Machines and the Nature of Knowledge, Harvard University Press, Cambridge, Mass. (1994).

16 David B. Fogel, Evolutionary Computation : Toward a New Philosophy of Machine Intelligence, IEEE Press, ISBN: 0780310381, (1995).

17 J. Brown, C. Heydemann, J. Soukup, "Theater Air Command and Control Simulation Facility ABL Test 6," Airborne Laser Program report, Phillips Laboratory, Albuquerque NM, (1994)

18 The Extended Air Defense Simulation (EADSIM) User's Reference Manual, US Army SSDC, Huntsville Alabama (1993)

19 W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes in C, Cambridge University Press, (1992).

20 P. D. Stroud, "Anisoplanatism in adaptive optics compensation of a focused beam with use of distributed beacons," J. Opt. Soc. Am. A / Vol. 13, No 4, pp 868-874, April 1996.

21 P. D. Stroud, "Statistics of intermediate duration averages of atmospheric scintillation," Optical Engineering, Vol. 35, No. 2, pp 543-548, Feb 1996.

22 J. H. Holland, Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, MA, (1992).

23 D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Co., (1989).

24 Handbook of intelligent control: neural, fuzzy, and adaptive approaches, D. A. White and D. A. Sofge, eds., Van Nostrand Reinhold, New York, (1992).